Specifying & Refining a File Transfer Protocol

MEB: Chapter 4

EECS3342 Z: System
Specification and Refinement
Winter 2022

CHEN-WEI WANG

Learning Outcomes

This module is designed to help you review:

- What a ***Requirement Document*** (***RD***) is
- What a ***refinement*** is
- Writing **_formal_ _specifications_**
  - (Static) <u>contexts</u>: constants, axioms, theorems
  - (Dynamic) <u>machines</u>: variables, invariants, events, guards, actions
- ***Proof Obligations*** (***POs***) associated with proving:
  - ***refinements***
  - system ***properties***
- Applying ***inference rules*** of the ***sequent calculus***

A Different Application Domain

- The bridge controller we *specified*, *refined*, and *proved* exemplifies a *reactive system*, working with the physical world via:
  - *sensors*   [a, b, c, ml_pass, il_pass]
  - *actuators*   [ml_tl, il_tl]

- We now study an example exemplifying a *distributed program* :
  - A *protocol* followed by two *agents*, residing on **distinct** geographical locations, on a computer **network**
  - Each file is transmitted *asynchronously*:
    bytes of the file do **not** arrive at the *receiver* all at one go.
  - Language of *predicates*, *sets*, and *relations* required
  - The **same** principles of generating *proof obligations* apply.

Requirements Document:
File Transfer Protocol (FTP)

LASSONDE
SCHOOL OF ENGINEERING

You are required to implement a system for transmitting files between *agents* over a computer network.

Page Source: https://www.venafi.com

Each *R-Description* is an **atomic** *specification* of an intended *functionality* or a desired *property* of the working system.

REQ1	The protocol ensures the copy of a file from the sender to the receiver.

REQ2	The file is supposed to be made of a sequence of items.

REQ3	The file is sent piece by piece between the two sites.

Refinement Strategy

- Recall the *__design strategy of progressive refinements__*.
    - **0.** *__initial__ __model__* (m_0): a file is transmitted from the *__sender__* to the *__receiver__*. [**REQ1**]

      However, at this *__most abstract__* model:
        - file transmitted from *__sender__* to *__receiver__* **synchronously** & **instantaneously**
        - transmission process *__abstracted__* away
    - **1.** *__1st refinement__* (m_1 *__refining__* m_0):

      transmission is done *__asynchronously__*   [**REQ2**, **REQ3**]

      However, at this <u>more concrete</u> model:
        - **<u>no</u>** communication between *__sender__* and *__receiver__*
        - exchanges of *__messages__* and *__acknowledgements__* *__abstracted__* away
    - **2.** *__2nd refinement__* (m_2 *__refining__* m_1):

      communication mechanism <u>elaborated</u>   [**REQ2**, **REQ3**]
    - **3.** *__final, 3rd refinement__* (m_3 *__refining__* m_2):

      communication mechanism <u>optimized</u>   [**REQ2**, **REQ3**]

- Recall   *Correct by Construction* :

    From each *__model__* to its *__refinement__*, only a <u>manageable</u> amount of details
    are added, making it *__feasible__* to conduct **analysis** and **proofs**.

Model m_0: Abstraction

- In this <u>most</u> **_abstract_** perception of the protocol, we do **not** consider the **_sender_** and **_receiver_**:
  - residing in geographically distinct locations
  - communicating via message exchanges
- Instead, we focus on this single **_requirement_**:

REQ1	The protocol ensures the copy of a file from the sender to the receiver.

- **_Abstraction Strategy_** :
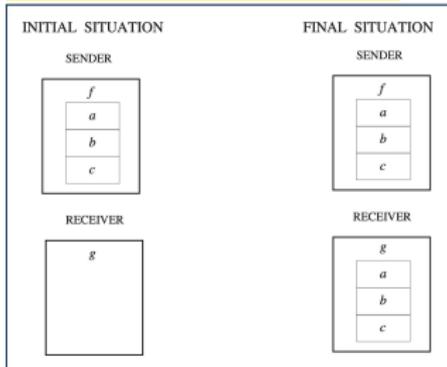
- Observe the system with the **_process of transmission abstracted_** away

- **<u>only</u>** meant to inform **_what_** the protocol is supposed to achieve

- **<u>not</u>** meant to detail **_how_** the transmission is achieved

Math Background Review

Refer to LECTURE 1 for reviewing:
- Predicates   [e.g., ∀]
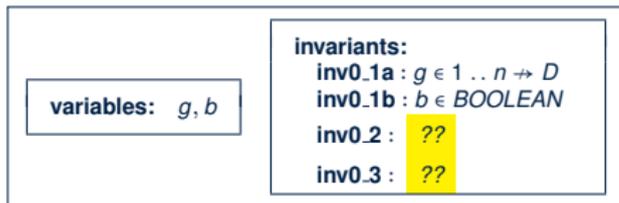- Sets
- Relations and Operations
- Functions

Model m_0: **Abstract State Space**

1. The **<u>static</u>** part formulates the *file* (from the *sender*'s end)

   as a sequence of data items:

sets: D, *BOOLEAN*	constants: n, f

   axioms:
   axm0_1 : $n > 0$
   axm0_2 : $f \in 1 \mathinner{.\,.} n \to D$
   axm0_3 : $BOOLEAN = \{TRUE, FALSE\}$

2. The **<u>dynamic</u>** part of the state consists of two *variables*:

   ✓ *g*: file from the *receiver*'s end

   ✓ *b*: whether or not the *transmission* is completed

variables: g, b	**invariants:** **inv0_1a** : $g \in 1 \mathinner{.\,.} n \nrightarrow D$ **inv0_1b** : $b \in BOOLEAN$ **inv0_2** : ?? **inv0_3** : ??

   ✓ **inv0_1a** and **inv0_1b** are *typing* constraints.

   ✓ **inv0_2** specifies what happens **<u>before</u>** the transmission

   ✓ **inv0_3** specifies what happens **<u>after</u>** the transmission

Model m_0: State Transitions via Events

- The system acts as an **ABSTRACT STATE MACHINE (ASM)** : it *evolves* as *actions of enabled events* change values of variables, subject to *invariants*.

- Initially, before the transmission:

```
init
    begin
        ??
    end
```

  ○ Nothing has been transmitted to the *receiver*.

  ○ The *transmission* process has not been completed.

- Finally, after the transmission:

```
final
    when
        ??
    then
        ??
    end
```

  ○ The entire file *f* has been transmitted to the *receiver*.

  ○ The *transmission* process has been completed.

  ○ In this *abstract* model:

    • Think of the transmission being *instantaneous*.
    • A later *refinement* specifies how *f* is transmitted *asynchronously*.

PO of Invariant Establishment

- How many **sequents** to be proved?   [# invariants]

- We have <u>four</u> **sequents** generated for **event** *init* of model m_0:

1.
$$n > 0$$
$$f \in 1 .. n \rightarrow D$$
$$BOOLEAN = \{ TRUE, FALSE \}$$
$$\vdash$$
$$\varnothing \in 1 .. n \nrightarrow D$$
     init/inv0_1a/INV

2.
$$n > 0$$
$$f \in 1 .. n \rightarrow D$$
$$BOOLEAN = \{ TRUE, FALSE \}$$
$$\vdash$$
$$FALSE \in BOOLEAN$$
     init/inv0_1b/INV

3.
$$n > 0$$
$$f \in 1 .. n \rightarrow D$$
$$BOOLEAN = \{ TRUE, FALSE \}$$
$$\vdash$$
$$FALSE = FALSE \Rightarrow \varnothing = \varnothing$$
     init/inv0_2/INV

4.
$$n > 0$$
$$f \in 1 .. n \rightarrow D$$
$$BOOLEAN = \{ TRUE, FALSE \}$$
$$\vdash$$
$$FALSE = TRUE \Rightarrow \varnothing = f$$
     init/inv0_3/INV

- <u>Exercises</u>: Prove the above sequents related to **invariant establishment**.

PO of Invariant Preservation

- How many **sequents** to be proved?   [# non-init events × # invariants]

- We have <u>four</u> **sequents** generated for **event** *final* of model m_0:

<div style="border:1px solid black; padding:4px; display:inline-block">

$n > 0$
$f \in 1 .. n \rightarrow D$
$BOOLEAN = \{ TRUE, FALSE \}$
$g \in 1 .. n \nrightarrow D$
$b \in BOOLEAN$
$b = FALSE \Rightarrow g = \varnothing$
$b = TRUE \Rightarrow g = f$
$b = FALSE$
$\vdash$
$f \in 1 .. n \nrightarrow D$

</div>

final/inv0_1a/INV

<div style="border:1px solid black; padding:4px; display:inline-block">

$n > 0$
$f \in 1 .. n \rightarrow D$
$BOOLEAN = \{ TRUE, FALSE \}$
$g \in 1 .. n \nrightarrow D$
$b \in BOOLEAN$
$b = FALSE \Rightarrow g = \varnothing$
$b = TRUE \Rightarrow g = f$
$b = FALSE$
$\vdash$
$TRUE \in BOOLEAN$

</div>

final/inv0_1b/INV

<div style="border:1px solid black; padding:4px; display:inline-block">

$n > 0$
$f \in 1 .. n \rightarrow D$
$BOOLEAN = \{ TRUE, FALSE \}$
$g \in 1 .. n \nrightarrow D$
$b \in BOOLEAN$
$b = FALSE \Rightarrow g = \varnothing$
$b = TRUE \Rightarrow g = f$
$b = FALSE$
$\vdash$
$TRUE = FALSE \Rightarrow f = \varnothing$

</div>

final/inv0_2/INV

<div style="border:1px solid black; padding:4px; display:inline-block">

$n > 0$
$f \in 1 .. n \rightarrow D$
$BOOLEAN = \{ TRUE, FALSE \}$
$g \in 1 .. n \nrightarrow D$
$b \in BOOLEAN$
$b = FALSE \Rightarrow g = \varnothing$
$b = TRUE \Rightarrow g = f$
$b = FALSE$
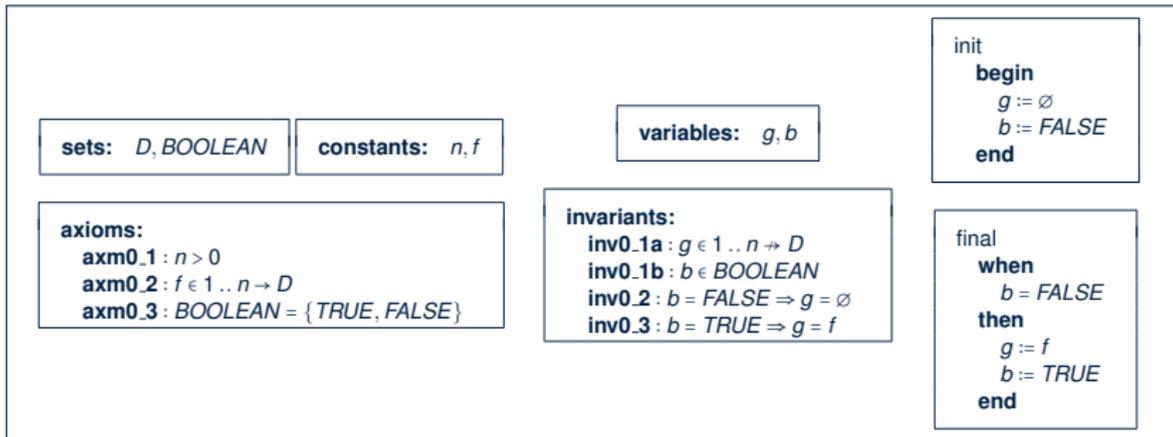$\vdash$
$TRUE = TRUE \Rightarrow f = f$

</div>

final/inv0_3/INV

- <u>Exercises</u>: Prove the above sequents related to **invariant preservation**.

Initial Model: Summary

- Our **_initial model_** m_0 is **_provably correct_** w.r.t.:
  - Establishment of **_Invariants_**
  - Preservation of **_Invariants_**
  - **_Deadlock_** Freedom   [EXERCISE]
- Here is the **_specification_** of m_0:

sets: $D, BOOLEAN$	**constants:** n, f

variables: g, b

init
  begin
    $g := \varnothing$
    $b := FALSE$
  end

axioms:
  axm0_1 : $n > 0$
  axm0_2 : $f \in 1 .. n \rightarrow D$
  axm0_3 : $BOOLEAN = \{ TRUE, FALSE \}$

invariants:
  inv0_1a : $g \in 1 .. n \nrightarrow D$
  inv0_1b : $b \in BOOLEAN$
  inv0_2 : $b = FALSE \Rightarrow g = \varnothing$
  inv0_3 : $b = TRUE \Rightarrow g = f$

final
  when
    $b = FALSE$
  then
 $g := f$
    $b := TRUE$
  end

Model m_1: "More Concrete" Abstraction

- In m_0, the transmission (evt. `final`) is **synchronous** and **instantaneous**.
- The 1st **refinement** has a more **concrete** perception of the file transmission:
  - The sender's file is coped <u>gradually</u>, **element by element**, to the receiver.
    - → Such progress is denoted by occurrences of a **new event** `receive`.

h: elements transmitted so far

r: index of element to be sent

abstract variable ***g*** is replaced by ***concrete*** variables ***h*** and ***r***.
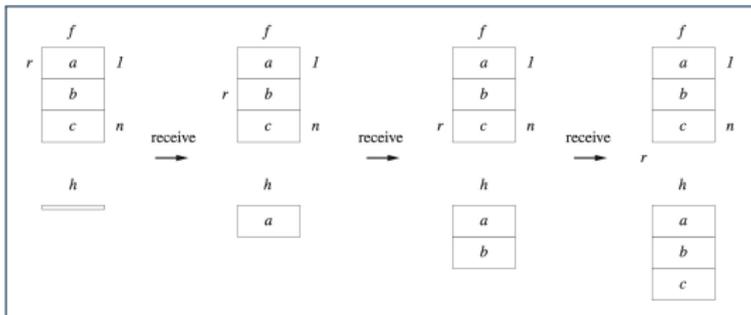
  - Nonetheless, communication between two agents remain **abstracted** away!
- That is, we focus on these two **intended functionalities**:

REQ2	The file is supposed to be made of a sequence of items.
REQ3	The file is sent piece by piece between the two sites.

- We are **obliged to prove** this **added concreteness** is **consistent** with m_0.

Model m_1: Refined, Concrete State Space

1. The **static** part remains the same as **m_0**:

| sets: D, BOOLEAN | constants: n, f |

axioms:
axm0_1 : $n > 0$
axm0_2 : $f \in 1 .. n \to D$
axm0_3 : $BOOLEAN = \{ TRUE, FALSE \}$

2. The **dynamic** part formulates the *gradual* transmission process:

| **variables:** b, h, r | **invariants:**
 inv1_1 : $r \in 1 .. n + 1$
 inv1_2 : *??*
 inv1_3 : *??*
 thm1_1 : *??* |

⋄ **inv1_1**: typing constraint

⋄ **inv2_2**: elements up to index *r - 1* have been transmitted

⋄ **inv2_3**: transmission completed **means** no more elements to be transmitted

⋄ **thm1_1**: transmission completed **means** receiver has a complete copy of sender's file

⋄ A *theorem*, once proved as *derivable from invariants*, needs **not** be proved for *preservation* by events.

- To prove that a **theorem** can be derived from the **invariants**:

variables: b, h, r	**invariants:** $inv1\_1$: $r \in 1 .. n + 1$ $inv1\_2$: $h = (1 .. r - 1) \lhd f$ $inv1\_3$: $b = TRUE \Rightarrow r = n + 1$ $thm1\_1$: $b = TRUE \Rightarrow h = f$

- We need to prove the following **sequent**:

  $$r \in 1 .. n + 1$$
  $$h = (1 .. r - 1) \lhd f$$
  $$b = TRUE \Rightarrow r = n + 1$$
  $$\vdash$$
  $$b = TRUE \Rightarrow h = f$$

- <u>Exercise</u>: Prove the above sequent.

- Initially, <u>before</u> the transmission:

  ```
  init
    begin
      ??
    end
  ```

  ◇ The ***transmission*** process has not been completed.

  ◇ Nothing has been transmitted to the ***receiver***.

  ◇ First file element is available for transmission.

- While the transmission is <u>ongoing</u>:

  ```
  receive
    when
      ??
    then
      ??
    end
  ```

  ◇ **While** sender has **more** file elements available for transmission:
    - Next file element is received and ***accumulated*** to the receiver's copy.
    - Sender's ***next available*** file element is updated.

  ◇ In this ***concrete*** model:
    - Receiver having access to sender's private variable *r* is ***unrealistic***.
    - A later ***refinement*** specifies how two agents communicate.

- Finally, <u>after</u> the transmission:

  ```
  final
    when
      ??
    then
      ??
    end
  ```

  ◇ **When** sender has **no** more file element available for transmission:
    - The ***transmission*** process is marked as completed.

PO of Invariant Establishment

- How many **sequents** to be proved?   [# invariants]

- We have <u>three</u> **sequents** generated for **event** *init* of model m_1:

1.	$n > 0$ $f \in 1 .. n \to D$ $BOOLEAN = \{TRUE, FALSE\}$ $\vdash$ $1 \in 1 .. n + 1$	**init/inv1_1/INV**

2.	$n > 0$ $f \in 1 .. n \to D$ $BOOLEAN = \{TRUE, FALSE\}$ $\vdash$ $\varnothing \in (1 .. 1 - 1) \lhd f$	**init/inv1_2/INV**

3.	$n > 0$ $f \in 1 .. n \to D$ $BOOLEAN = \{TRUE, FALSE\}$ $\vdash$ $FALSE = TRUE \Rightarrow 1 = n + 1$	**init/inv1_3/INV**

- <u>Exercises</u>: Prove the above sequents related to **invariant establishment**.

- We have <u>three</u> *sequents* generated for *old event* final of model m_1.
- Here is one of the sequents:

$n > 0$
$f \in 1 .. n \to D$
$BOOLEAN = \{TRUE, FALSE\}$
$g \in 1 .. n \nrightarrow D$
$b \in BOOLEAN$
$b = FALSE \Rightarrow g = \varnothing$
$b = TRUE \Rightarrow g = f$
$r \in 1 .. n + 1$
$h = (1 .. r - 1) \lhd f$
$b = TRUE \Rightarrow r = n + 1$
$b = FALSE$
$r = n + 1$
$\vdash$
$r \in 1 .. n + 1$

final/inv1_1/INV

- <u>Exercises</u>: Formulate & prove other sequents of *invariant preservation*.

- We have <u>three</u> *sequents* generated for *new event* *receive* of model m_1:

receive/inv1_1/INV

$n > 0$
$f \in 1 .. n \to D$
$BOOLEAN = \{TRUE, FALSE\}$
$g \in 1 .. n \nrightarrow D$
$b \in BOOLEAN$
$b = FALSE \Rightarrow g = \varnothing$
$b = TRUE \Rightarrow g = f$
$r \in 1 .. n + 1$
$h = (1 .. r - 1) \lhd f$
$b = TRUE \Rightarrow r = n + 1$
$r \leq n$
$\vdash$
$(r + 1) \in 1 .. n + 1$

receive/inv1_2/INV

$n > 0$
$f \in 1 .. n \to D$
$BOOLEAN = \{TRUE, FALSE\}$
$g \in 1 .. n \nrightarrow D$
$b \in BOOLEAN$
$b = FALSE \Rightarrow g = \varnothing$
$b = TRUE \Rightarrow g = f$
$r \in 1 .. n + 1$
$h = (1 .. r - 1) \lhd f$
$b = TRUE \Rightarrow r = n + 1$
$r \leq n$
$\vdash$
$h \cup \{(r, f(r))\} = (1 .. (r + 1) - 1) \lhd f$

receive/inv1_3/INV

$n > 0$
$f \in 1 .. n \to D$
$BOOLEAN = \{TRUE, FALSE\}$
$g \in 1 .. n \nrightarrow D$
$b \in BOOLEAN$
$b = FALSE \Rightarrow g = \varnothing$
$b = TRUE \Rightarrow g = f$
$r \in 1 .. n + 1$
$h = (1 .. r - 1) \lhd f$
$b = TRUE \Rightarrow r = n + 1$
$r \leq n$
$\vdash$
$b = TRUE \Rightarrow (r + 1) = n + 1$
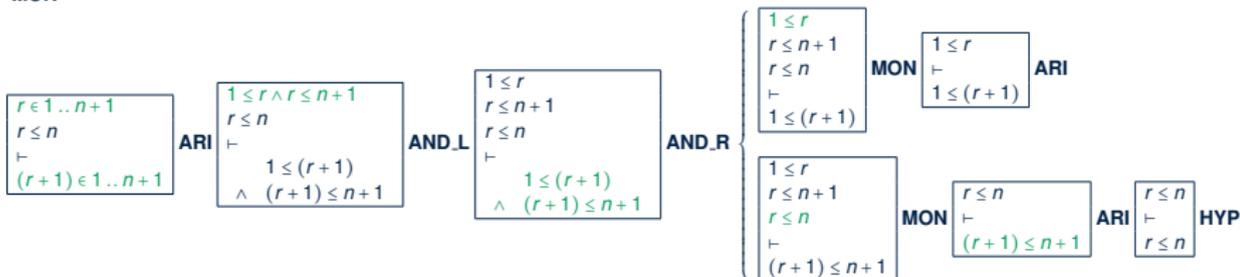
- <u>Exercises</u>: Prove the above sequents of *invariant preservation*.

Proving Refinement: receive/inv1_1/INV

$n > 0$
$f \in 1 .. n \rightarrow D$
$BOOLEAN = \{ TRUE, FALSE \}$
$g \in 1 .. n \nrightarrow D$
$b \in BOOLEAN$
$b = FALSE \Rightarrow g = \varnothing$
$b = TRUE \Rightarrow g = f$
$r \in 1 .. n + 1$
$h = (1 .. r - 1) \vartriangleleft f$
$b = TRUE \Rightarrow r = n + 1$
$r \leq n$
$\vdash$
$(r + 1) \in 1 .. n + 1$

MON

$r \in 1 .. n + 1$
$r \leq n$
$\vdash$
$(r + 1) \in 1 .. n + 1$

ARI

$1 \leq r \wedge r \leq n + 1$
$r \leq n$
$\vdash$
$\quad 1 \leq (r + 1)$
$\wedge \quad (r + 1) \leq n + 1$

AND_L

$1 \leq r$
$r \leq n + 1$
$r \leq n$
$\vdash$
$\quad 1 \leq (r + 1)$
$\wedge \quad (r + 1) \leq n + 1$

AND_R

$1 \leq r$
$r \leq n + 1$
$r \leq n$
$\vdash$
$1 \leq (r + 1)$

MON

$1 \leq r$
$\vdash$
$1 \leq (r + 1)$

ARI

$1 \leq r$
$r \leq n + 1$
$r \leq n$
$\vdash$
$(r + 1) \leq n + 1$

MON

$r \leq n$
$\vdash$
$(r + 1) \leq n + 1$

ARI

$r \leq n$
$\vdash$
$r \leq n$

HYP

$n > 0$
$f \in 1 .. n \to D$
$BOOLEAN = \{TRUE, FALSE\}$
$g \in 1 .. n \to D$
$b \in BOOLEAN$
$b = FALSE \Rightarrow g = \varnothing$
$b = TRUE \Rightarrow g = f$
$r \in 1 .. n + 1$
$h = (1 .. r - 1) \lhd f$
$b = TRUE \Rightarrow r = n + 1$
$r \leq n$
$\vdash$
$h \cup \{(r, f(r))\} = (1 .. (r + 1) - 1) \lhd f$

MON

$f \in 1 .. n \to D$
$r \in 1 .. n + 1$
$h = (1 .. r - 1) \lhd f$
$r \leq n$
$\vdash$
$h \cup \{(r, f(r))\} = (1 .. (r + 1) - 1) \lhd f$

ARI

$f \in 1 .. n \to D$
$1 \leq r$
$h = (1 .. r - 1) \lhd f$
$r \leq n$
$\vdash$
$h \cup \{(r, f(r))\} = (1 .. (r + 1) - 1) \lhd f$

**EQ_LR,
MON,
ARI**

$f \in 1 .. n \to D$
$1 \leq r$
$r \leq n$
$\vdash$
$(1 .. r - 1) \lhd f \cup \{(r, f(r))\} = (1 .. r) \lhd f$

ARI

LASSONDE
SCHOOL OF ENGINEERING

$$n > 0$$
$$f \in 1 .. n \to D$$
$$BOOLEAN = \{TRUE, FALSE\}$$
$$g \in 1 .. n \nrightarrow D$$
$$b \in BOOLEAN$$
$$b = FALSE \Rightarrow g = \varnothing$$
$$b = TRUE \Rightarrow g = f$$
$$r \in 1 .. n + 1$$
$$h = (1 .. r - 1) \lhd f$$
$$b = TRUE \Rightarrow r = n + 1$$
$$r \le n$$
$$\vdash$$
$$b = TRUE \Rightarrow (r + 1) = n + 1$$

MON

| $b = TRUE \Rightarrow r = n + 1$ $r \le n$ $\vdash$ $b = TRUE \Rightarrow (r + 1) = n + 1$ | **IMP_R** | $b = TRUE \Rightarrow r = n + 1$ $r \le n$ $b = TRUE$ $\vdash$ $(r + 1) = n + 1$ | **IMP_L** | $r = n + 1$ $r \le n$ $b = TRUE$ $\vdash$ $(r + 1) = n + 1$ | **EQ_LR, MON** | $n + 1 \le n$ $b = TRUE$ $\vdash$ $((n + 1) + 1) = n + 1$ | **ARI, MON** | $\bot$ $\vdash$ $((n + 1) + 1) = n + 1$ | **FALSE_L** |

- Recall:
  - Interleaving of **new** events charactered as an integer expression: **variant**.
  - A variant $V(c, w)$ may refer to constants and/or **concrete** variables.
  - For m_1, let's try $\boxed{\textbf{variants} : n + 1 - r}$
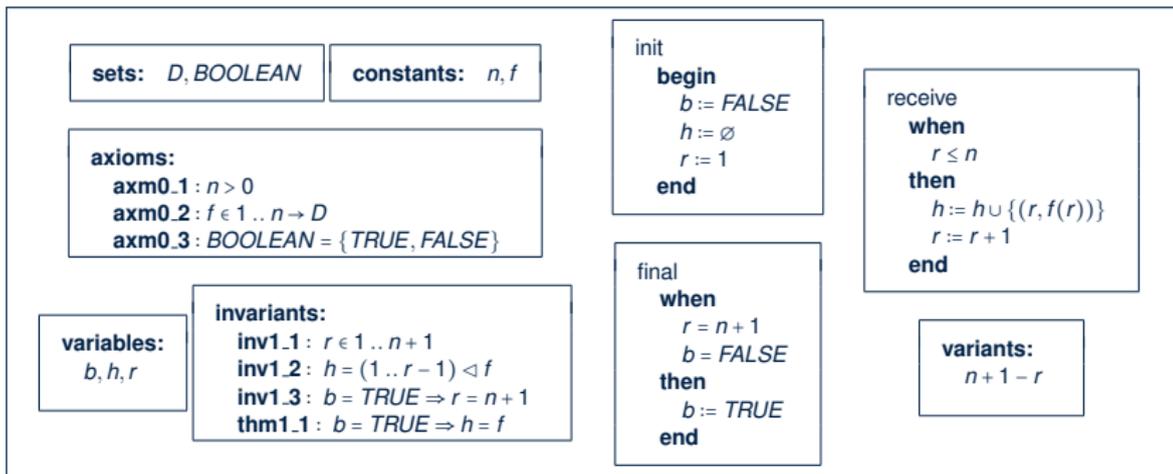
- Accordingly, for the **new** event *receive*:

$$
\begin{array}{l}
n > 0 \\
f \in 1 .. n \to D \\
BOOLEAN = \{TRUE, FALSE\} \\
g \in 1 .. n \nrightarrow D \\
b \in BOOLEAN \\
b = FALSE \Rightarrow g = \varnothing \\
b = TRUE \Rightarrow g = f \\
r \in 1 .. n + 1 \\
h = (1 .. r - 1) \vartriangleleft f \\
b = TRUE \Rightarrow r = n + 1 \\
r \leq n \\
\vdash \\
n + 1 - (r + 1) < n + 1 - r
\end{array}
\qquad \text{receive/VAR}
$$

Exercises: Prove **receive/VAR** and Formulate/Prove **receive/NAT**.

First Refinement: Summary

- The **first refinement** m_1 is **provably correct** w.r.t.:
  - Establishment of **Concrete Invariants** [*init*]
  - Preservation of **Concrete Invariants** [old & new events]
  - Strengthening of **guards** [old events, EXERCISE]
  - **Convergence** (a.k.a. livelock freedom, non-divergence) [new events, EXERCISE]
  - Relative **Deadlock** Freedom [EXERCISE]
- Here is the **specification** of m_1:

<div>

sets: D, *BOOLEAN*   **constants:** n, f

axioms:
axm0_1 : $n > 0$
axm0_2 : $f \in 1 .. n \to D$
axm0_3 : *BOOLEAN* $= \{ TRUE, FALSE \}$

variables: b, h, r

invariants:
inv1_1 : $r \in 1 .. n + 1$
inv1_2 : $h = (1 .. r - 1) \lhd f$
inv1_3 : $b = TRUE \Rightarrow r = n + 1$
thm1_1 : $b = TRUE \Rightarrow h = f$

init
  begin
    $b := FALSE$
    $h := \varnothing$
    $r := 1$
  end

final
  when
    $r = n + 1$
    $b = FALSE$
  then
    $b := TRUE$
  end

receive
  when
    $r \leq n$
  then
    $h := h \cup \{ (r, f(r)) \}$
    $r := r + 1$
  end

variants:
  $n + 1 - r$

</div>

Learning Outcomes

A Different Application Domain

Requirements Document:
File Transfer Protocol (FTP)

Requirements Document: R-Descriptions

Refinement Strategy

Model m_0: Abstraction

Math Background Review

Model m_0: Abstract State Space

Model m_0: State Transitions via Events

PO of Invariant Establishment